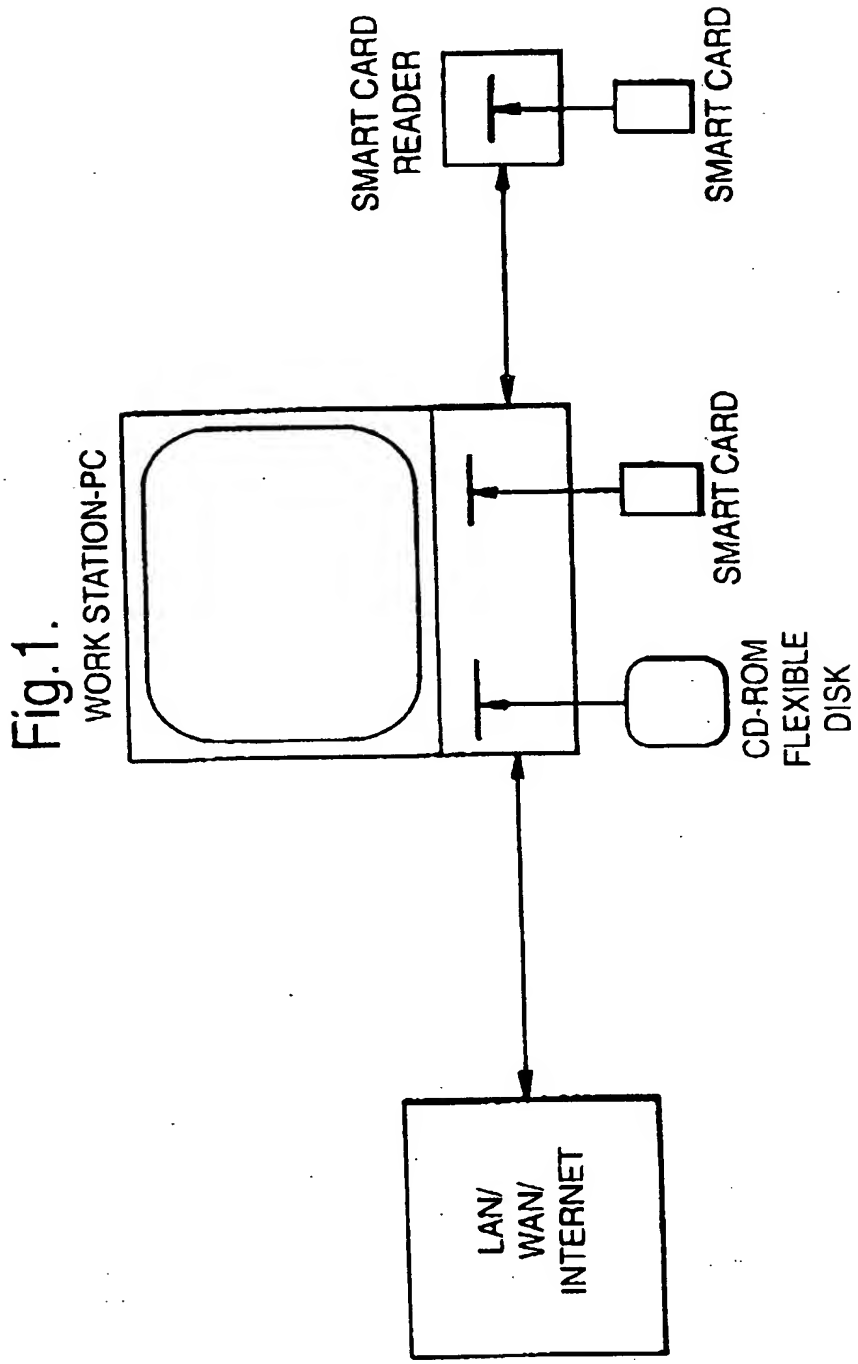

SMART CARD READER
SMART CARD
WORK STATION-PC
SMART CARD
CD-ROM FLEXIBLE DISK
LAN/ WAN/ INTERNET

Fig.1.

Fig.2.

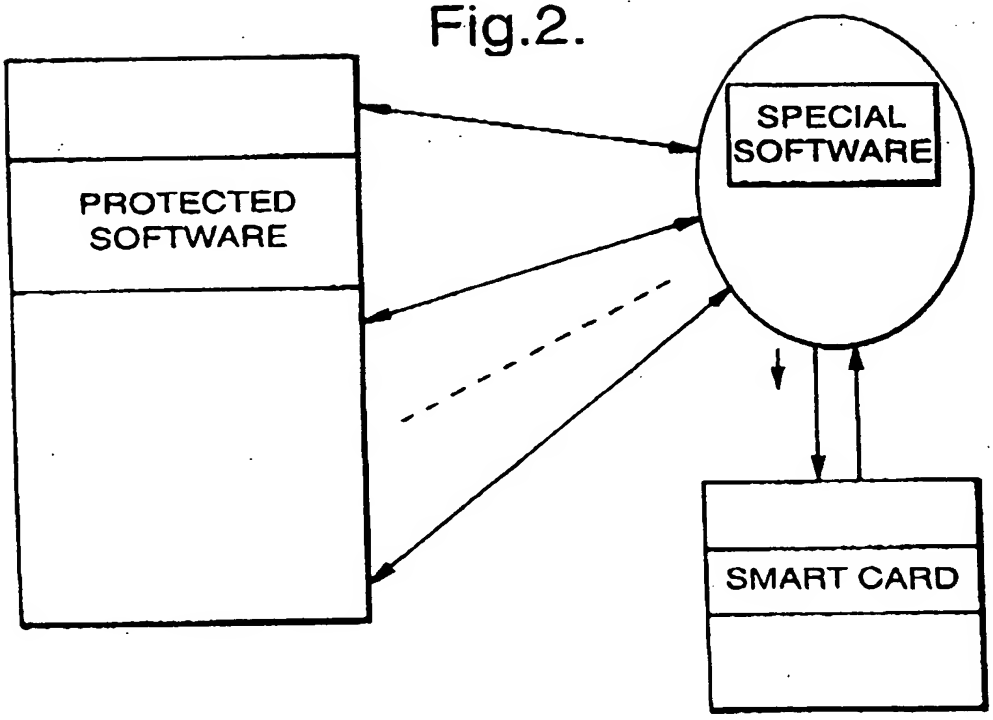
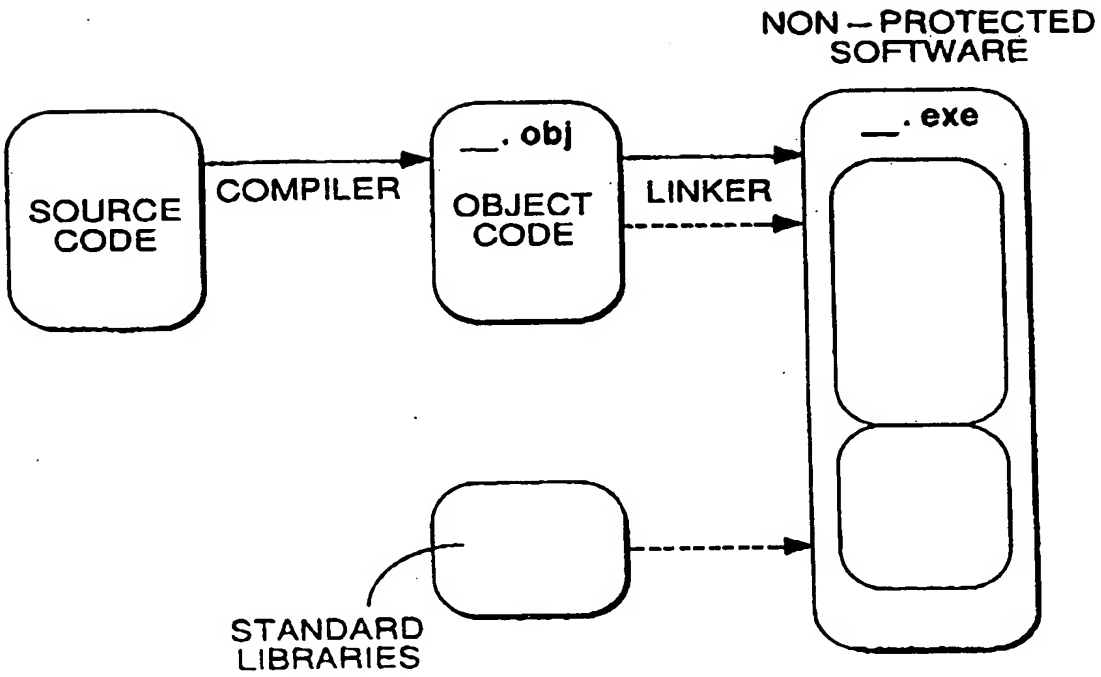


Fig.3.


NON – PROTECTED SOFTWARE
SOURCE CODE
COMPILER
__. obj
OBJECT CODE
LINKER
__. exe
STANDARD LIBRARIES

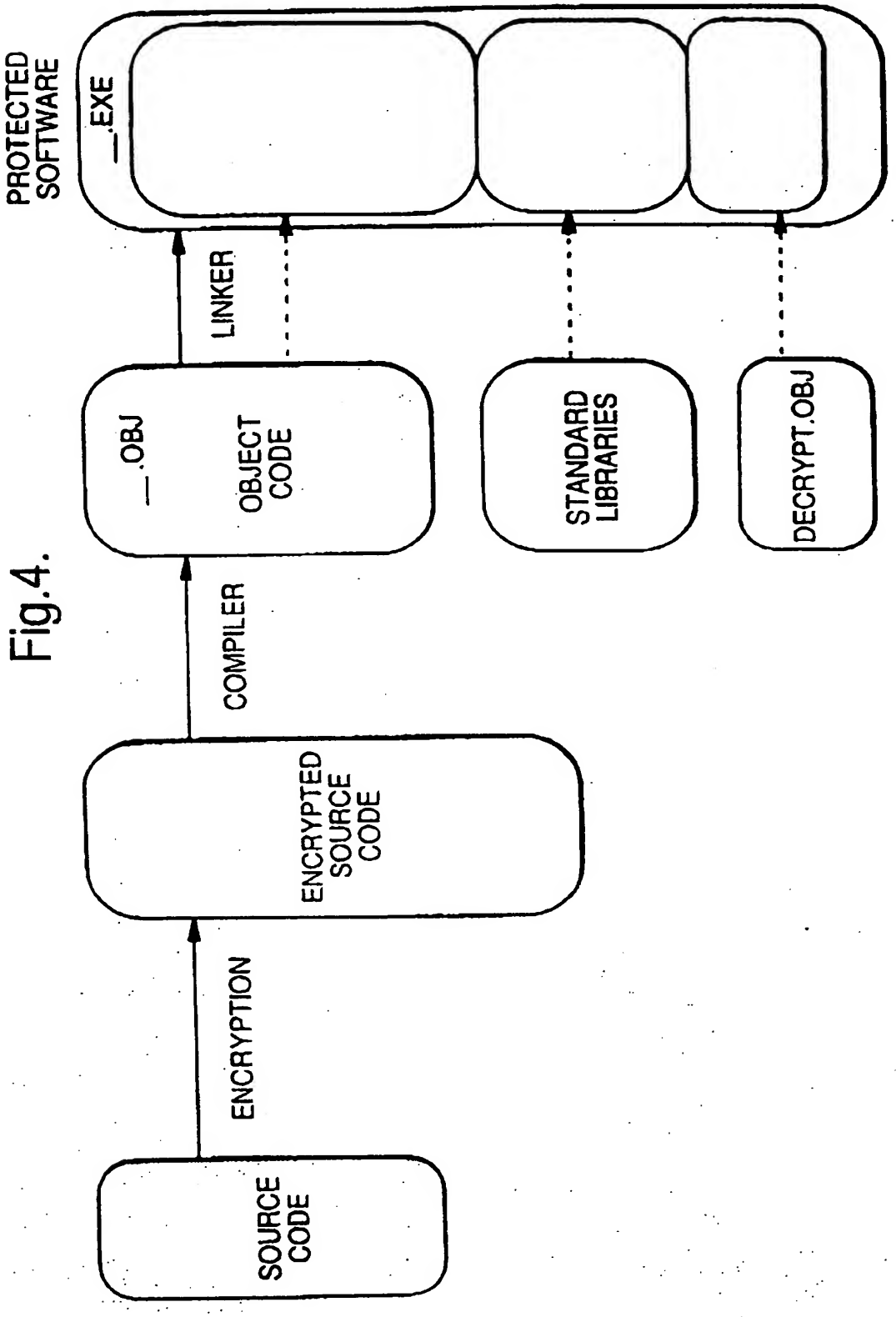
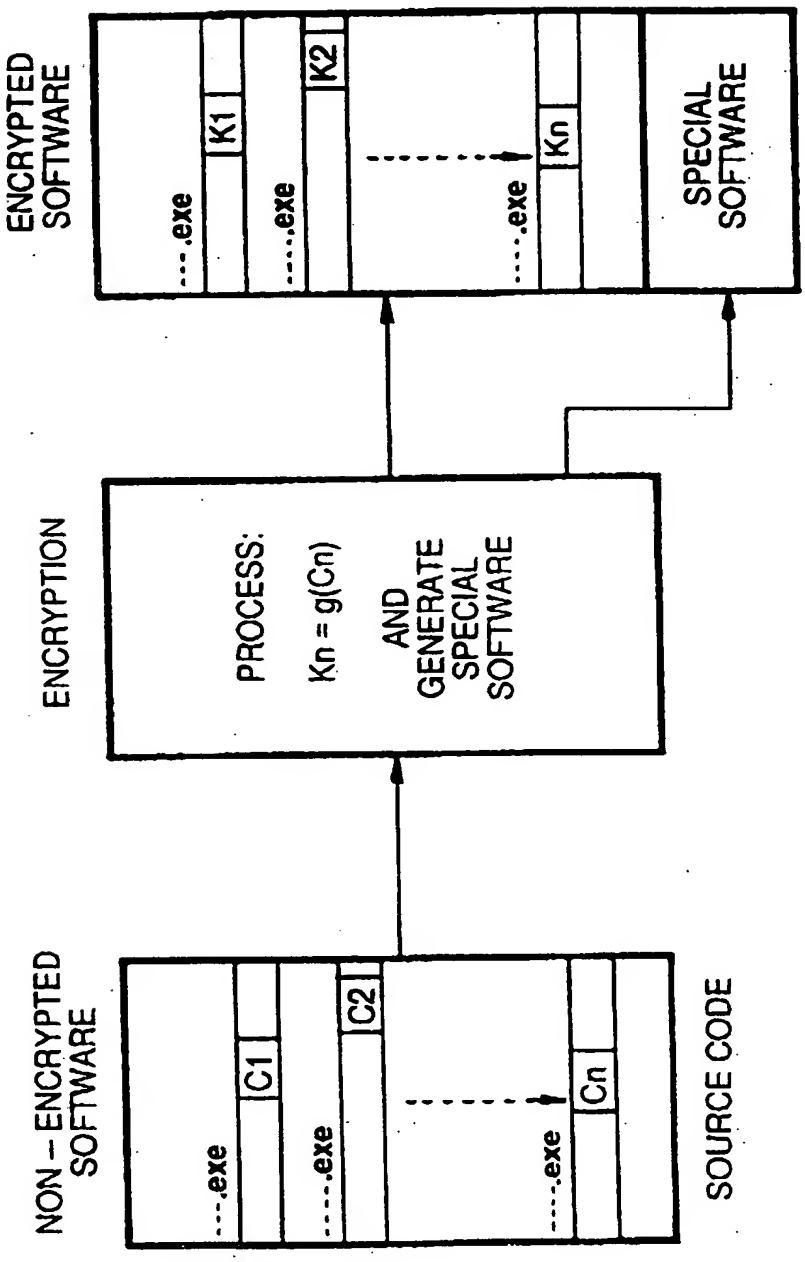

PROTECTED SOFTWARE
_.EXE
LINKER
_.OBJ
OBJECT CODE
STANDARD LIBRARIES
DECRYPT.OBJ
COMPILER
ENCRYPTED SOURCE CODE
ENCRYPTION
SOURCE CODE

Fig.4.

Fig.5.


ENCRYPTED SOFTWARE
---.exe
K1
----.exe
K2
----.exe
Kn
SPECIAL SOFTWARE
ENCRYPTION
PROCESS:
Kn = g(Cn)
AND GENERATE SPECIAL SOFTWARE
NON – ENCRYPTED SOFTWARE
---.exe
C1
----.exe
C2
----.exe
Cn
SOURCE CODE

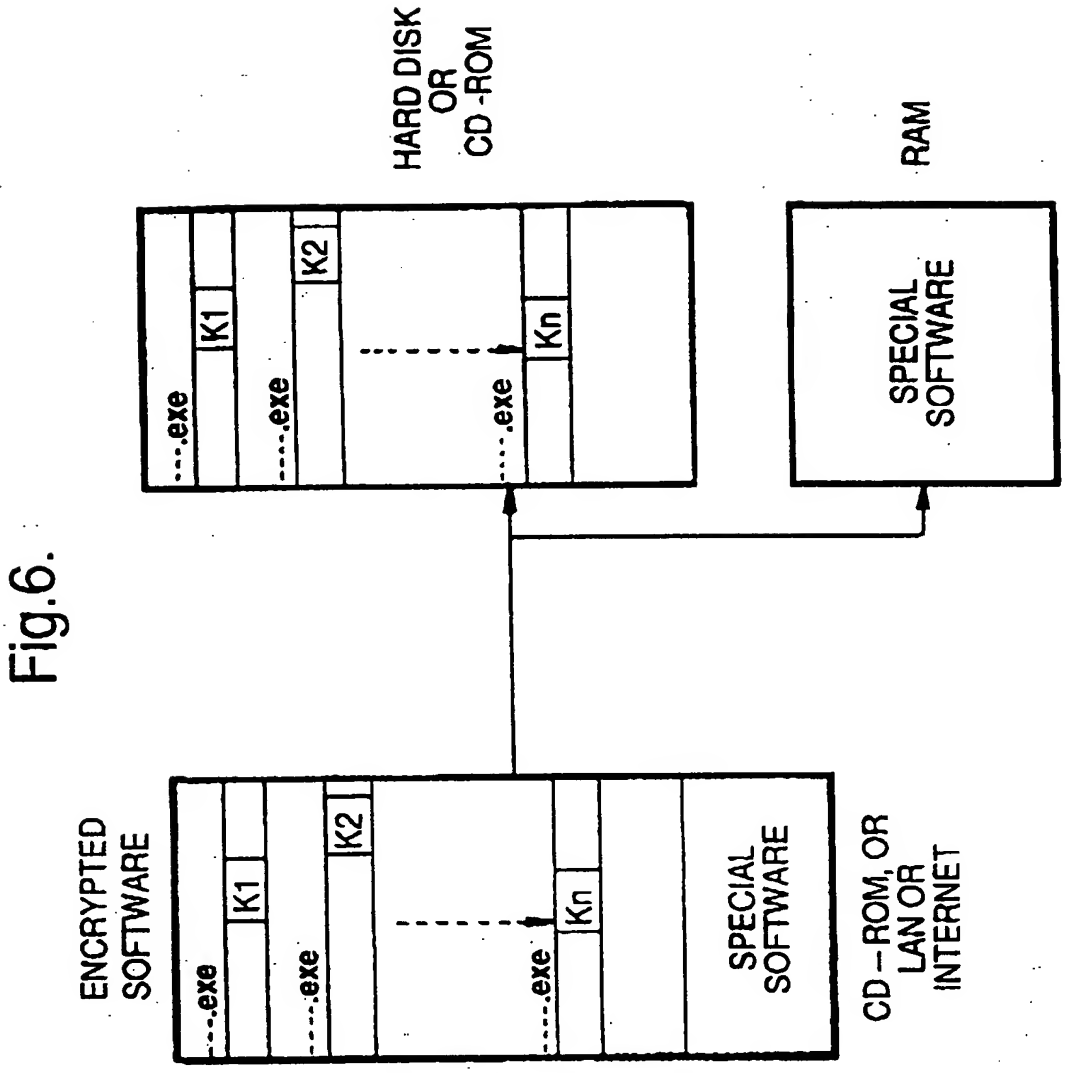

HARD DISK OR CD-ROM
RAM
K1
K2
Kn
.exe
SPECIAL SOFTWARE
ENCRYPTED SOFTWARE
CD-ROM, OR LAN OR INTERNET

Fig.6.

Fig.7.

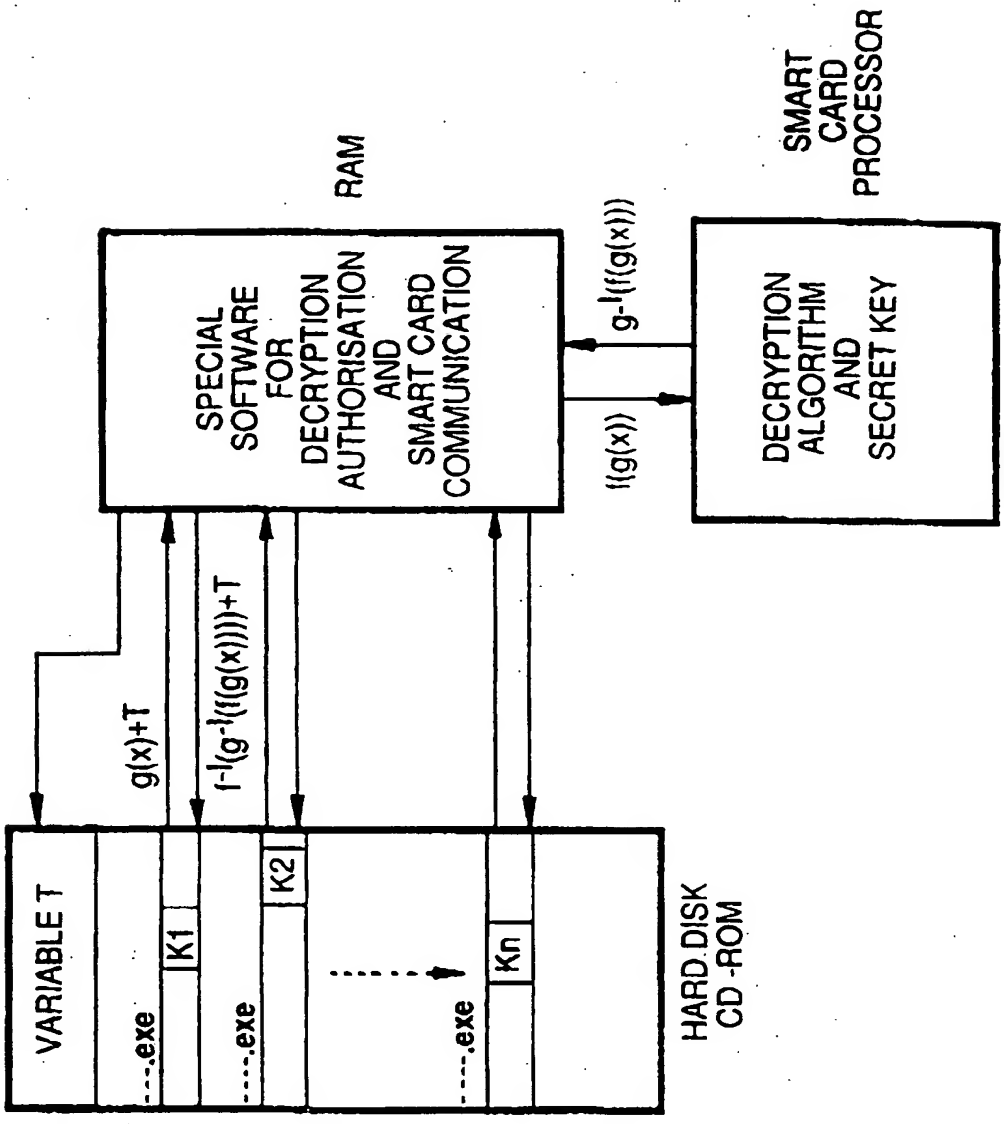

RAM
SMART CARD PROCESSOR
SPECIAL SOFTWARE FOR DECRYPTION AUTHORISATION AND SMART CARD COMMUNICATION
DECRYPTION ALGORITHM AND SECRET KEY
g-1(f(g(x)))
f(g(x))
g(x)+T
f-1(g-1(f(g(x))))+T
VARIABLE T
K1
K2
Kn
.exe
.exe
.exe
HARD DISK CD-ROM

Fig.8.

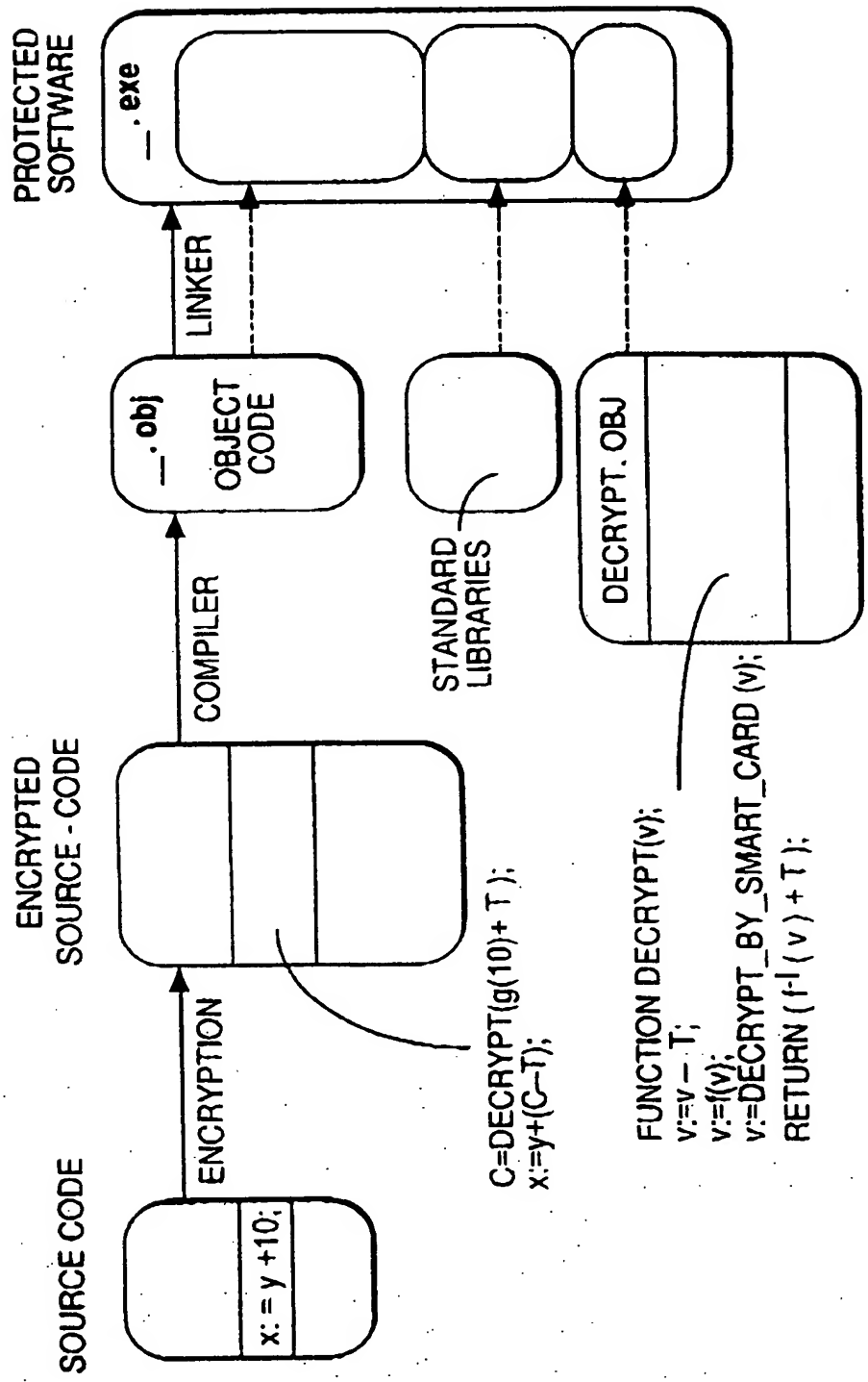

PROTECTED SOFTWARE
_ . exe
LINKER
_ . obj
OBJECT CODE
COMPILER
STANDARD LIBRARIES
DECRYPT. OBJ
ENCRYPTED SOURCE - CODE
ENCRYPTION
SOURCE CODE
x: = y +10;
C=DECRYPT(g(10)+ T);
x:=y+(C–T);
FUNCTION DECRYPT(v);
v:=v – T;
v:=f(v);
v:=DECRYPT_BY_SMART_CARD (v);
RETURN (f-1 (v) + T);

Fig.9.

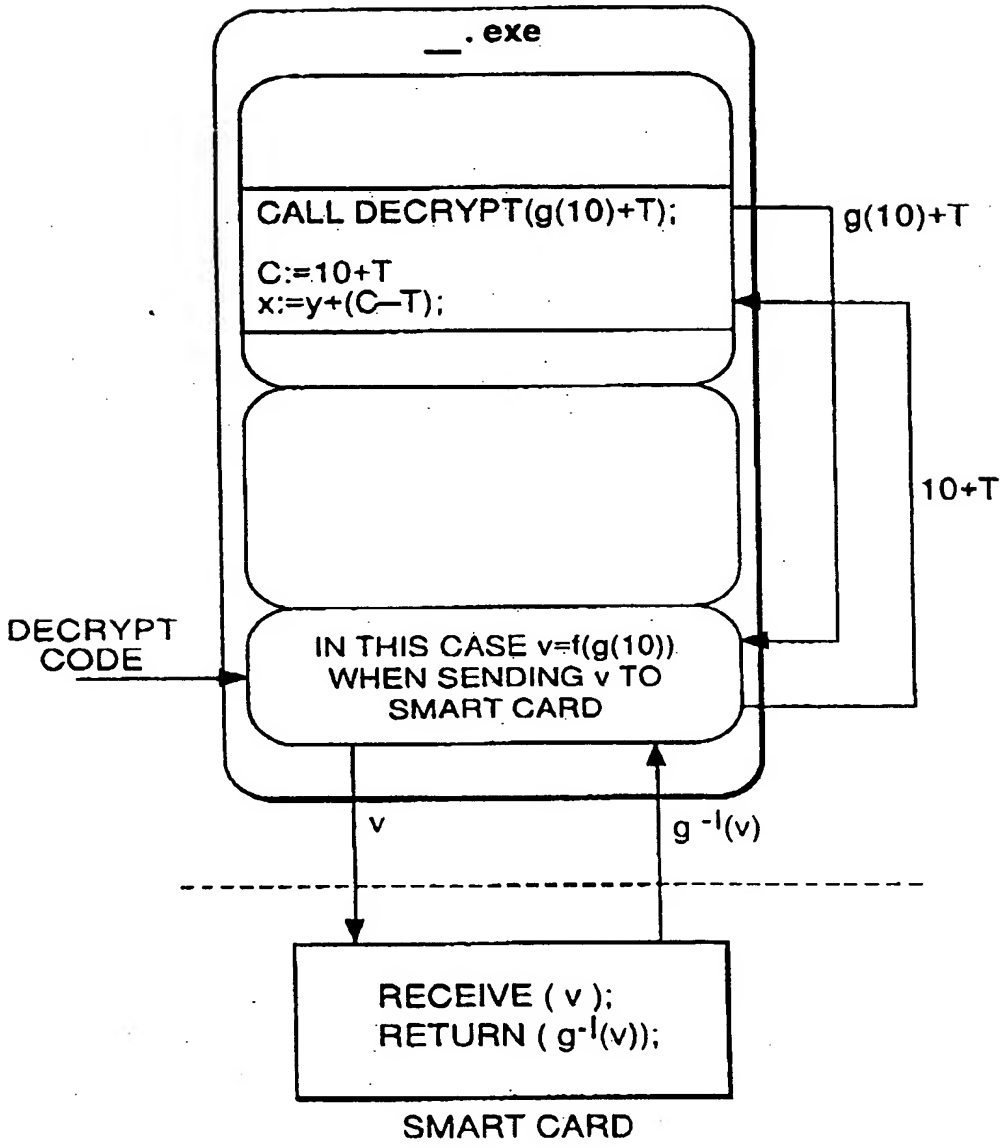
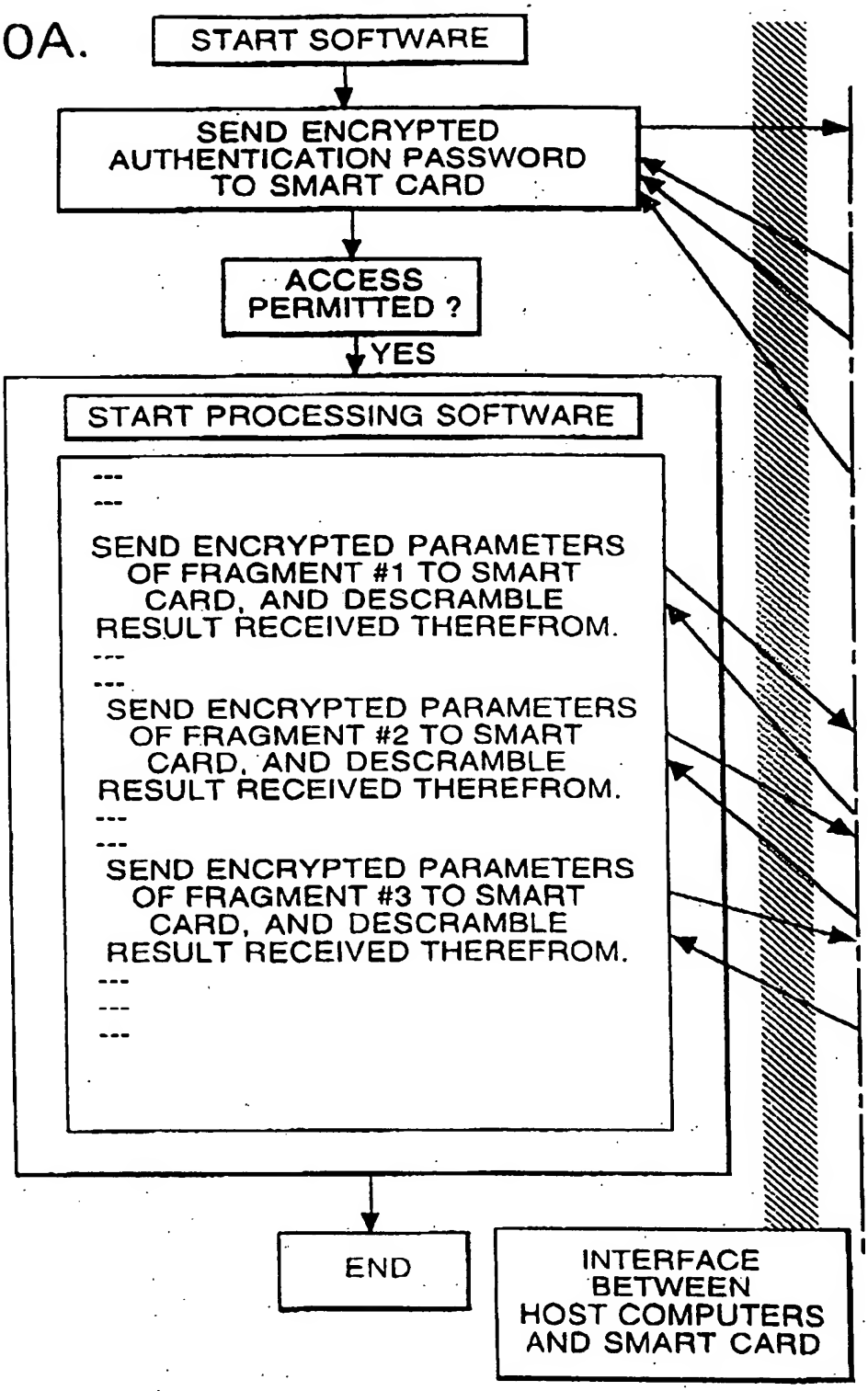

__. exe
CALL DECRYPT(g(10)+T);
C:=10+T
x:=y+(C—T);
g(10)+T
10+T
DECRYPT CODE
IN THIS CASE v=f(g(10)) WHEN SENDING v TO SMART CARD
v
g -1(v)
RECEIVE (v);
RETURN (g-1(v));
SMART CARD

Fig.10A.


START SOFTWARE
SEND ENCRYPTED AUTHENTICATION PASSWORD TO SMART CARD
ACCESS PERMITTED ?
YES
START PROCESSING SOFTWARE

SEND ENCRYPTED PARAMETERS OF FRAGMENT #1 TO SMART CARD, AND DESCRAMBLE RESULT RECEIVED THEREFROM.

SEND ENCRYPTED PARAMETERS OF FRAGMENT #2 TO SMART CARD, AND DESCRAMBLE RESULT RECEIVED THEREFROM.

SEND ENCRYPTED PARAMETERS OF FRAGMENT #3 TO SMART CARD, AND DESCRAMBLE RESULT RECEIVED THEREFROM.

END
INTERFACE BETWEEN HOST COMPUTERS AND SMART CARD

Fig.10B.

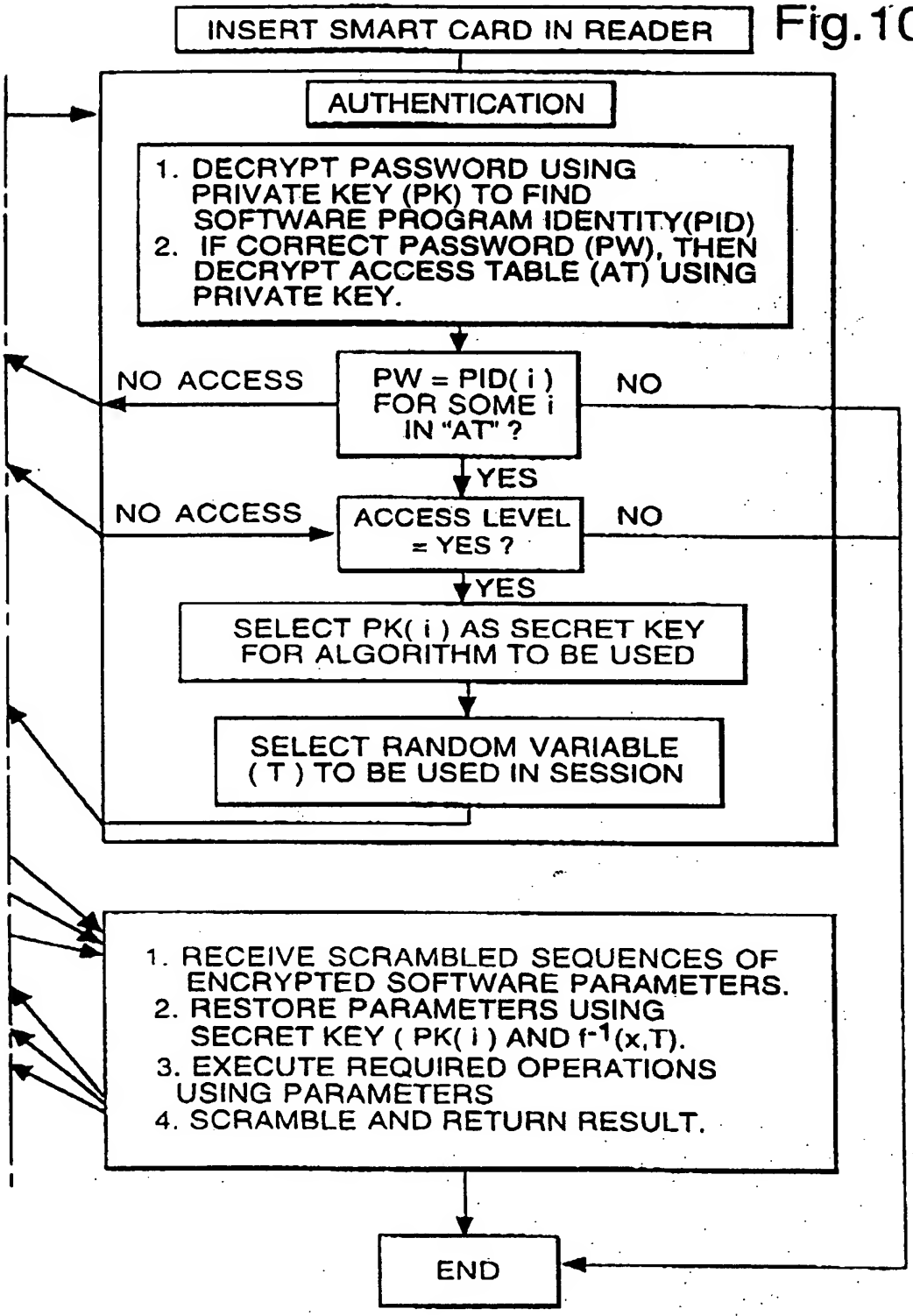
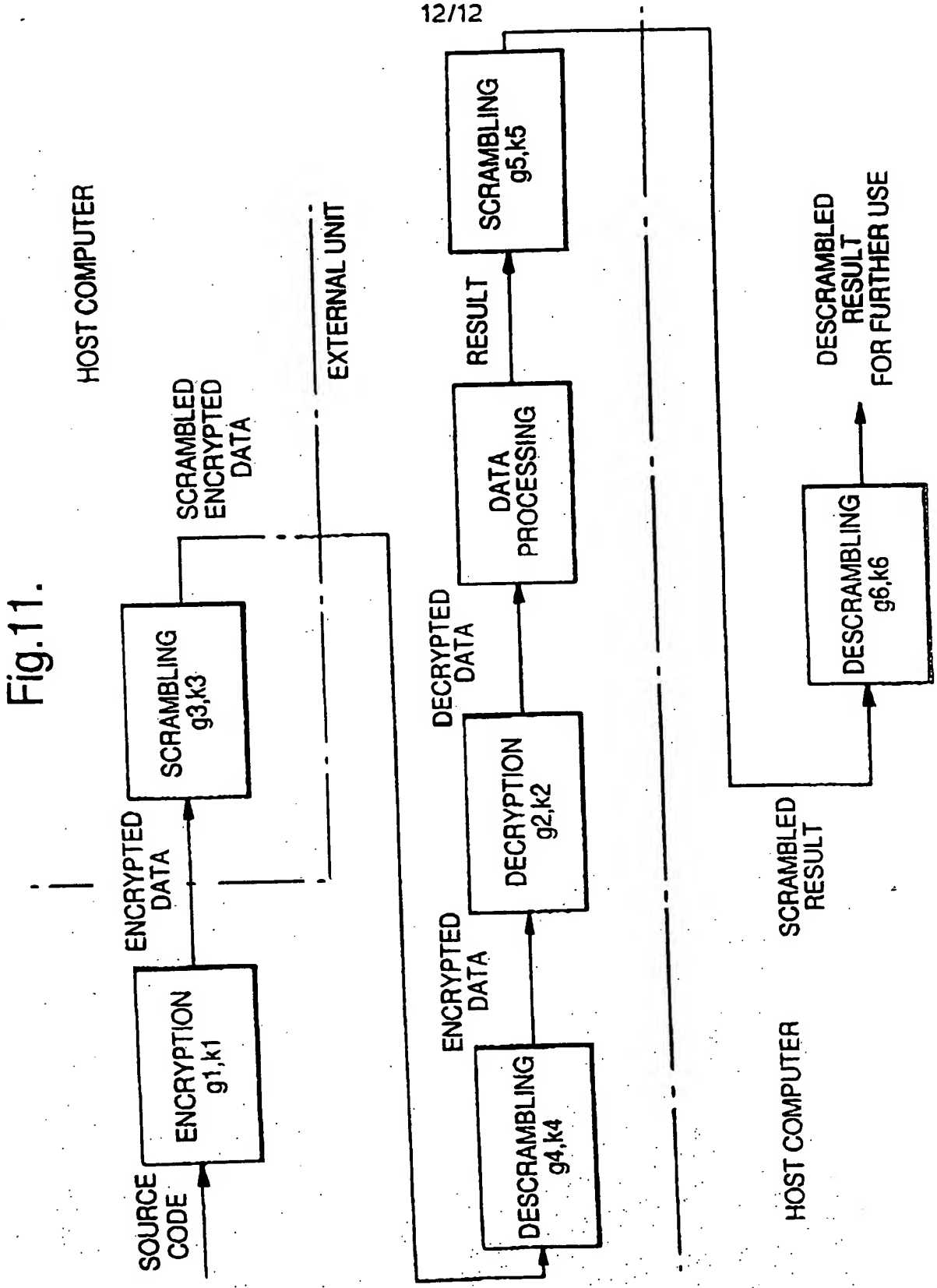

INSERT SMART CARD IN READER
AUTHENTICATION
1. DECRYPT PASSWORD USING PRIVATE KEY (PK) TO FIND SOFTWARE PROGRAM IDENTITY(PID)
2. IF CORRECT PASSWORD (PW), THEN DECRYPT ACCESS TABLE (AT) USING PRIVATE KEY.
NO ACCESS
PW = PID(i) FOR SOME i IN "AT" ?
NO
YES
NO ACCESS
ACCESS LEVEL = YES ?
NO
YES
SELECT PK(i) AS SECRET KEY FOR ALGORITHM TO BE USED
SELECT RANDOM VARIABLE (T) TO BE USED IN SESSION
1. RECEIVE SCRAMBLED SEQUENCES OF ENCRYPTED SOFTWARE PARAMETERS.
2. RESTORE PARAMETERS USING SECRET KEY (PK(i) AND $f^{-1}(x,T)$.
3. EXECUTE REQUIRED OPERATIONS USING PARAMETERS
4. SCRAMBLE AND RETURN RESULT.
END

Fig.11.


HOST COMPUTER
EXTERNAL UNIT
SOURCE CODE
ENCRYPTION g1,k1
ENCRYPTED DATA
SCRAMBLING g3,k3
SCRAMBLED ENCRYPTED DATA
DESCRAMBLING g4,k4
ENCRYPTED DATA
DECRYPTION g2,k2
DECRYPTED DATA
DATA PROCESSING
RESULT
SCRAMBLING g5,k5
SCRAMBLED RESULT
HOST COMPUTER
DESCRAMBLING g6,k6
DESCRAMBLED RESULT FOR FURTHER USE